Building a Custom HIView

Session 116

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Building a Custom HIView

Bryan Prusha and Curt Rothert
HIToolbox Engineers

http://developer.apple.com/wwdc2005/

What You'll Learn

- Subclassing HIView
- Drawing
- Mouse Tracking
- Keyboard Navigation and Text Input
- Accessibility
- Drag and Drop
- Archiving
- HIFramework

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
 CFStringRef           inClassID,
 CFStringRef           inSuperclassID,
 OptionBits            inOptions,
 EventHandlerUPP       inEventHandler,
 UInt32                inNumEvents,
 const EventTypeSpec* inEventList,
 void*                 inConstructData,
 HIObjectClassRef*     outClassRef );
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
 CFStringRef           inClassID,
 CFStringRef           inSuperclassID,
 OptionBits            inOptions,
 EventHandlerUPP       inEventHandler,
 UInt32                inNumEvents,
 const EventTypeSpec* inEventList,
 void*                 inConstructData,
 HIObjectClassRef*     outClassRef );
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
  CFStringRef           inClassID,
  CFStringRef           inSuperclassID,
  OptionBits            inOptions,
  EventHandlerUPP       inEventHandler,
  UInt32                inNumEvents,
  const EventTypeSpec*  inEventList,
  void*                 inConstructData,
  HIObjectClassRef*     outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
  CFStringRef           inClassID,
  CFStringRef           inSuperclassID,
  OptionBits            inOptions,
  EventHandlerUPP       inEventHandler,
  UInt32                inNumEvents,
  const EventTypeSpec*  inEventList,
  void*                 inConstructData,
  HIObjectClassRef*     outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                        CFSTR(“com.myco.myview”),
                        kHIViewClassID,
OptionBits              inOptions,
EventHandlerUPP         inEventHandler,
UInt32                  inNumEvents,
const EventTypeSpec*    inEventList,
void*                   inConstructData,
HIObjectClassRef*       outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                        CFSTR(“com.myco.myview”),
                        kHIViewClassID,
OptionBits              inOptions,
EventHandlerUPP         inEventHandler,
UInt32                  inNumEvents,
const EventTypeSpec*    inEventList,
void*                   inConstructData,
HIObjectClassRef*       outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                        CFSTR(“com.myco.myview”),
                        kHIViewClassID,
OptionBits              inOptions,
EventHandlerUPP         inEventHandler,
UInt32                  inNumEvents,
const EventTypeSpec*    inEventList,
void*                   inConstructData,
HIObjectClassRef*       outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
const EventTypeSpec kMyViewEvents[] =
{{ kEventClassHIObject, kEventHIObjectConstruct },
 { kEventClassHIObject, kEventHIObjectDestruct },
 { kEventClassHIObject, kEventHIObjectInitialize },
 { kEventClassControl,  kEventControlDraw } };

OSStatus HIObjectRegisterSubclass(
                    CFSTR(“com.myco.myview”),
                    kHIViewClassID,
 OptionBits         inOptions,
 EventHandlerUPP    inEventHandler,
                    GetEventTypeCount( kMyViewEvents ),
                    kMyViewEvents,
 void*              inConstructData,
 HIObjectClassRef*  outClassRef );
```

```
class MyView :: View
{
};
```

Subclassing HIView

Register the subclass

```
const EventTypeSpec kMyViewEvents[] =
{{ kEventClassHIObject, kEventHIObjectConstruct },
 { kEventClassHIObject, kEventHIObjectDestruct },
 { kEventClassHIObject, kEventHIObjectInitialize },
 { kEventClassControl,  kEventControlDraw } };

OSStatus HIObjectRegisterSubclass(
                    CFSTR(“com.myco.myview”),
                    kHIViewClassID,
 OptionBits         inOptions,
 EventHandlerUPP    inEventHandler,
                    GetEventTypeCount( kMyViewEvents ),
                    kMyViewEvents,
 void*              inConstructData,
 HIObjectClassRef*  outClassRef );
```

```
class MyView :: View
{
    MyView();
    ~MyView();

    Initialize();
    Draw();
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                    CFSTR(“com.myco.myview”),
                    kHIViewClassID,
OptionBits          inOptions,
EventHandlerUPP     inEventHandler,
                    GetEventTypeCount( kMyViewEvents ),
                    kMyViewEvents,
void*               inConstructData,
HIObjectClassRef*   outClassRef );
```

```
class MyView :: View
{
    MyView();
    ~MyView();

    Initialize();
    Draw();
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                    CFSTR(“com.myco.myview”),
                    kHIViewClassID,
 OptionBits         inOptions,
                    MyViewEventHandler,
                    GetEventTypeCount( kMyViewEvents ),
                    kMyViewEvents,
 void*              inConstructData,
 HIObjectClassRef*  outClassRef );

OSStatus MyViewEventHandler(
 EventHandlerCallRef inCallRef,
 EventRef            inEvent,
 void*               inUserData )
{
 // handle events
}
```

```
class MyView :: View
{
    MyView();
    ~MyView();

    Initialize();
    Draw();
};
```

Subclassing HIView

Register the subclass

```
OSStatus HIObjectRegisterSubclass(
                    CFSTR(“com.myco.myview”),
                    kHIViewClassID,
 OptionBits         inOptions,
                    MyViewEventHandler,
                    GetEventTypeCount( kMyViewEvents ),
                    kMyViewEvents,
 void*              inConstructData,
 HIObjectClassRef*  outClassRef );

OSStatus MyViewEventHandler(
 EventHandlerCallRef inCallRef,
 EventRef            inEvent,
 void*               inUserData )
{
 // handle events
}
```

```
class MyView :: View
{
    MyView() {};
    ~MyView() {};

    Initialize();
    Draw();
};
```

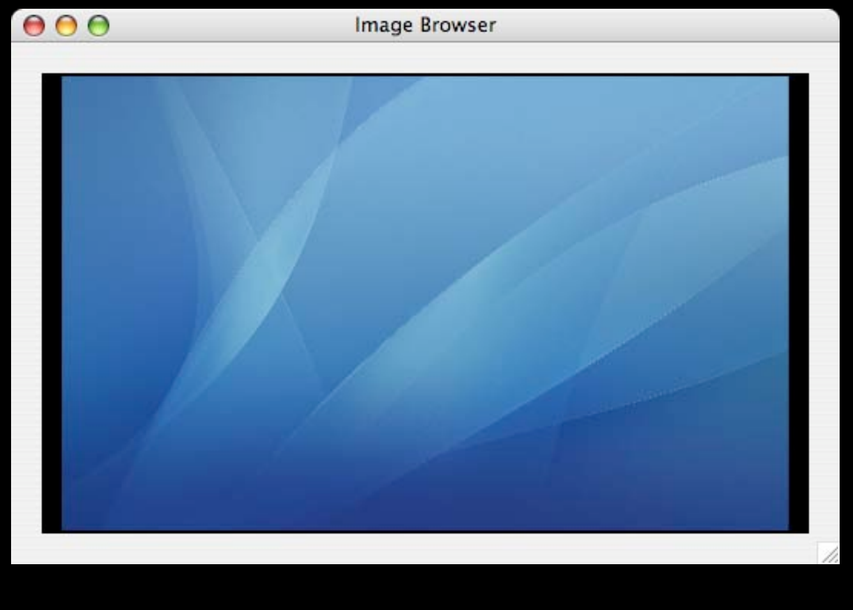
```
MyView::Initialize()
{
    // handle initialization
}

MyView::Draw()
{
    // handle drawing
}
```

Demo

http://developer.apple.com/wwdc2005/

Drawing

- kWindowCompositingAttribute
- Invalidation
  - HIViewSetNeedsDisplay
  - Invalidate on state change
    - kEventControlActivate
    - kEventControlDeactivate
    - kEventControlHiliteChanged
    - kEventControlEnabledStateChanged
- Drawing
  - kEventControlDraw



Demo

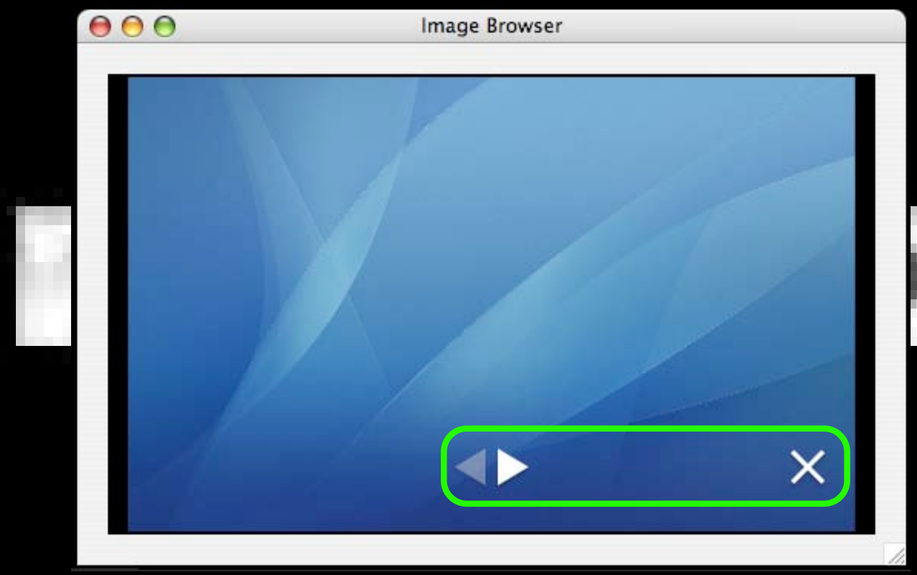
http://developer.apple.com/wwdc2005/

Mouse Tracking

- Hit Testing
  - kEventControlHitTest
    - Find the part of your view under the test point
- Basic Tracking
  - kEventControlHiliteChanged
  - kEventControlHit
    - Reacting to a mouse click
- Custom Tracking
  - kEventControlTrack
    - Handle a mouse down

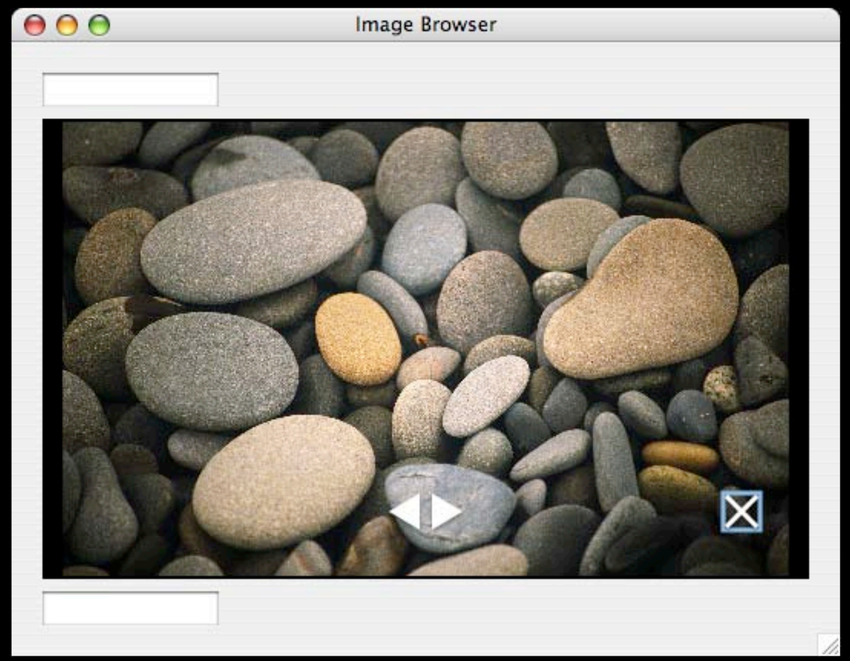
❌ Don't: while (StillDown()) …

✅ Do: TrackMouseLocation



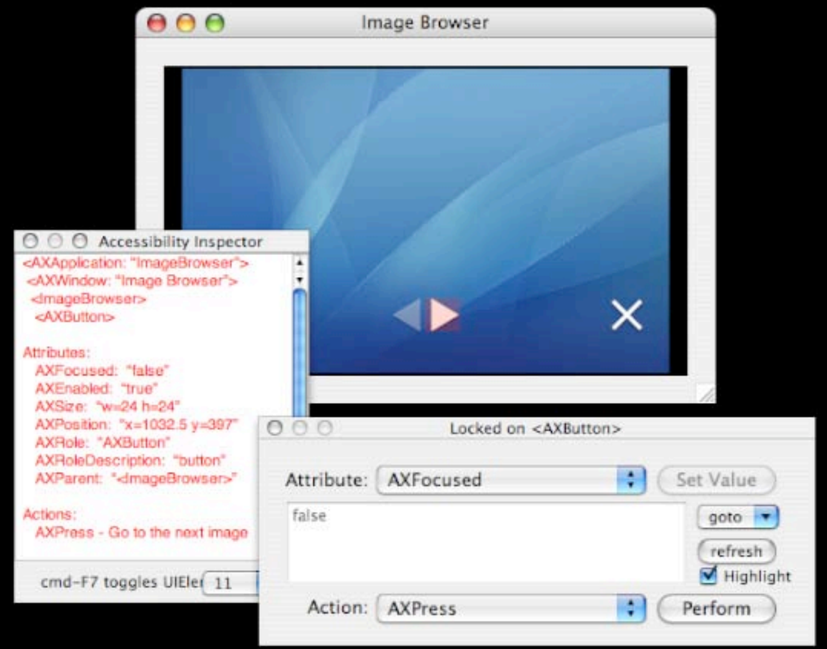
Keyboard Navigation and Text Input

- Navigation
  - kEventControlSetFocusPart
  - kEventControlGetFocusPart
  - kHIViewFeaturesGetsFocusOnClick
- Text Input
  - kEventClassTextInput
  - kEventTextInputUnicodeForKeyEvent
  - Avoid kEventRawKeyDown
  - HIViewSimulateClick



Accessibility

- kEventClassAccessibility
  - Children
    - Focused child, child at point
  - Attributes
    - Role, position, size, title, description
  - Actions
    - Press, increment, decrement, confirm, cancel
- Visit the Accessibility Lab on Friday, 2–5pm

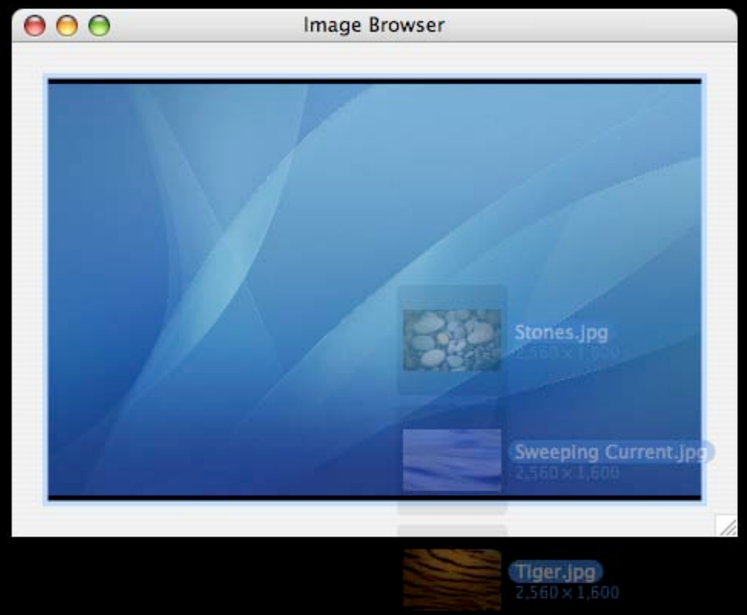


Demo

http://developer.apple.com/wwdc2005/

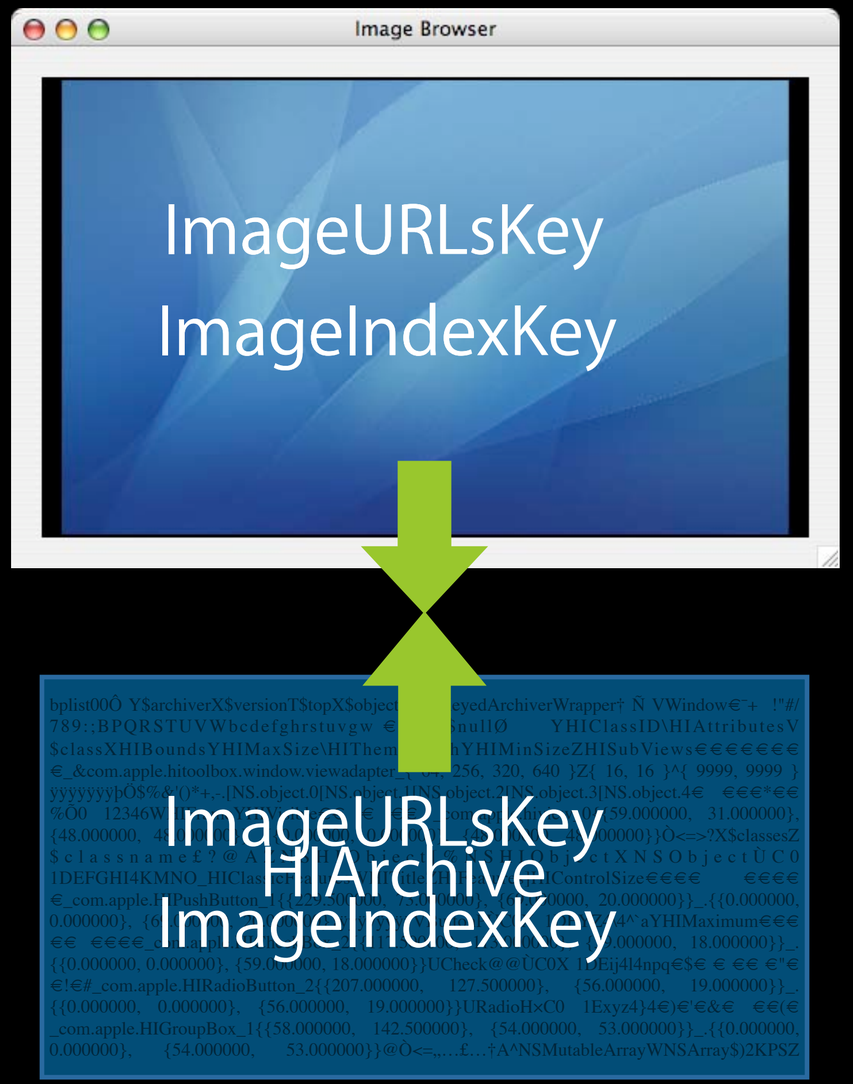
Drag and Drop

- Enabling Dragging
  - SetAutomaticControlDragTrackingEnabledForWindow
  - SetControlDragTrackingEnabled
- kEventDragEnter
  - kEventParamControlWouldAcceptDrop
- kEventDragWithin
- kEventDragLeave
- kEventDragReceive



Archiving

- Encoding
  - kEventHIObjectEncode
    - HIArchiveEncodeCFType
- Decoding
  - kEventHIObjectInitialize
    - HIArchiveCopyDecodedCFType
- HIObjectSetArchivingIgnored



Demo

http://developer.apple.com/wwdc2005/

HIFramework

- C++ framework
  - Wraps HIToolbox functionality
  - Simplifies HIObject subclassing
- HIFramework 1.1
  - TEventHandler >TObject >TView
  - Accessibility support

What's the Next Step?

- Experiment with the sample code
- Convert CDEFs to HIViews
- Support compositing

Who to Contact

HIToolbox Feedback
hitoolbox-feedback@group.apple.com

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Carbon and HIToolbox Feedback Forum	Session 131 Thursday 10:30am
Lab: Create a Custom HIView	All Week

Q&A

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference